# Exploring the Word

→ **Config Driven UI**

• A configuration is sent by backend which influences the UI or divides the view UI looks like.

→ **Why state variable is needed?**

Ans: Even if we change the normal JS variable it will get changed but since React doesn't tracks such variables so it any change there will be no nothing will be rerendered and hence on UI. changed
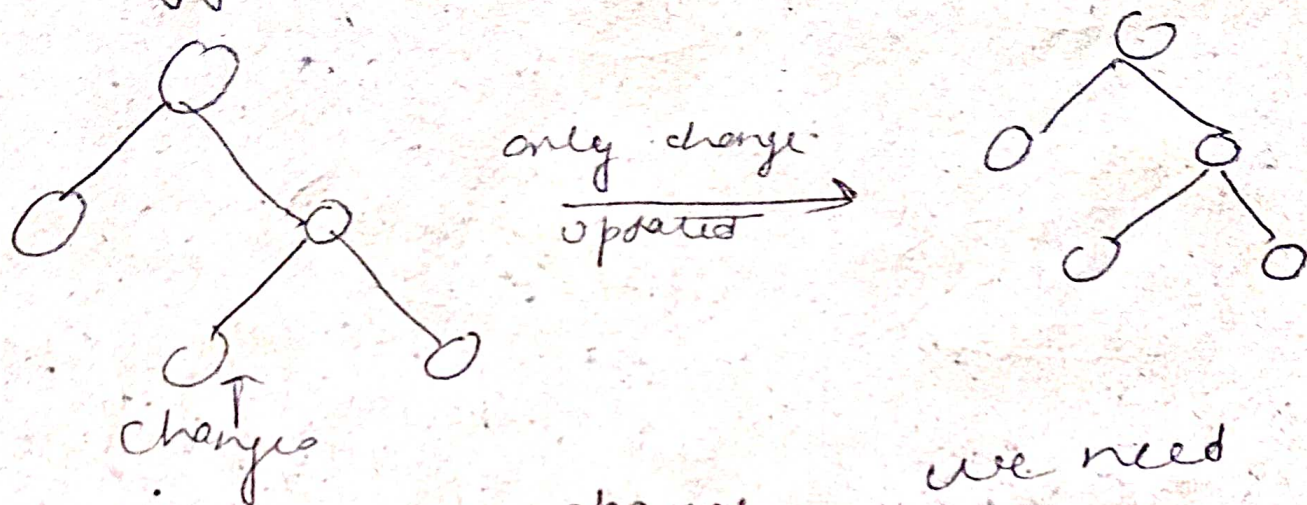
→ **Abstract Syntax Tree**

• Most expensive operation on UI is DOM manipulation.

→ React is fast only bez of faster DOM manipulation.

↓ bez of

Diff Algorithm
React Fiber
Reconcilliation Algorithm

→ Any state associated with component.
if state changes, whole component
gets rerendered and reconcilliation process
gets triggered.



only change
updated →

changes

→ for every UI change, we need
state variables.

Now, let's explore the world
making on calling API to call
the restaurants.

__Industry concept__

    ↳ microservices

__Monolithic__
• Every functionality will be on
  same project. (one project)
• Even if per min. changes
  we need to deploy the whole
  app
• lots of mvc, still run on this
  format only

# Microservices

( UI )          ( Backend API )          ( Notification )

( Log )          ( authenti cation )

- Everything on different port.
- ~~core~~ separation of concern.
- easier to ~~test~~ and modify.
- everything can be written in different tech stack.

**Q)** How all these are connected?

**Ans:** : 3000 ⟶ /
: 9 0 0 ⟶ / Log;
: 5000 ⟶ / notification

- different ports but same
  domain name.

→ Here, we are building only
  UI / microservices.

**Q** How to make API call in Javascript?

**Ans)** fetch → Browser function (window)

**Q** where should I make API call in React?

**Ans)** As in when my page loads, call an API and fills the data

two Approaches to do in React

1) Load → API → render page
   (300ms)   (500ms)

2) Load —timer→ API ⇒ [update] v2
        100ms    300ms      500ms

   good | user experience basically during the
   (we are | user) preferred way

   second hook

**→ Introducing use Effect HOOK**

↳ use Effect

**Syntax)** useEffect ( () => {

} , [ ] ) → callback firing
        ↑
   dependency array

→ decides when to call use Effect, not on every state changes

→ Two instances where component rerenders :—

1). either state changes or
2) props changes.

// empty dependency array ⇒ it will be called only once after initial render.

// dependency array [searchTerm] ⇒ once after initial render + everytime dependency changes

(Q) can we have multiple useEffect()?

→ Json, objects.

• Calls the swiggy public API.

(Q) Critical Rendering

→ Handling : Errors

• Making user Experience good
    ↳ Introduce shimmer UI
        ↓
    skeleton of the page UI

# Conditional Rendering

## Early Return

→ Difference b/w JS expression
& Statements.

# One can run only JS expression
inside JSX not statements.

→ making a login / Logout /
Authenticate user button.